ComBots Glossary/Formulas

Bot. One of the participants in a battle (also, "combatant")
Rogue. A "wild" Bot (i.e., a Bot that has no Meister)
Meister. A Bot's operator
Upgrade. When a Basic Bot changes into a Combat Bot
Awaken. When a Combat Bot changes into an Awakened Bot
Basic Bot. The lowest form of a Bot. The player's/teammates' Bots always start battle in this form. Opponents' Bots often but do not always start battle in this form.
Combat Bot. The second form of a Bot. Each Basic Bot has three Combat Bots that they can upgrade into during a battle, unlocked progressively as the Bot's Blueprint Experience increases.
Awakened Bot. The highest form of a Bot. Each Combat Bot has only one Awakened Bot that it can awaken into, unlocked after each of a Blueprint's Combat Bots is unlocked.
RAM. The resource required to install Software on a Bot. Stronger Software takes up more RAM, and a Bot's RAM increases as its Meister's Rank increases.
Load. The resource required to equip Hardware to a Bot. Stronger Hardware weighs more, taking up more Load, and a Bot's Load increases as its Meister's Rank increases
Bond. The level of friendship that the player has with a given teammate (between 0 and 3 Hearts). The higher the bond, the more options the player has to customize and command that teammate's Bot.
Software. "moves"/"attacks." Each piece of Software has a RAM cost, and a Bot must have enough RAM to install a given piece of Software.
Hardware. Equipment with five potential points of attachment (Internal, Headgear, Armor, Arm, and Boot). Basic Bots have three Internal slots, one Headgear slot, and two Arm slots; Combat and Awakened Bots have three Internal slots, one Headgear slot, two Arm slots, one Armor slot, and one Boot slot. Each piece of Hardware has a weight, and a Bot must have enough Load to handle the weight of all the Hardware equipped to it.
Ability. A Bot's passive ability. Each Bot has three Abilities to choose from, unlocked progressively as the Bot's Blueprint Experience increases. Only one Ability can be assigned to a Bot at a time
Rank. A Meister's level
Blueprint. The identity of a Bot, which determines all of its potential forms and, thus, also determines its stats, Abilities, and compatible Software.
Challenger NPC. An NPC that will challenge the player to a battle (as opposed to a Rogue)
Teammate. One of two members of the player's party besides the player themself. Eventually, a player will have up to 12 different teammates to choose from.
Special Battle. a battle against a unique NPC that is preprogrammed and connected to the story.
Endurance. A Bot's HP
Energy. A Bot's individual resource, a la Mana, that is consumed to use most pieces of Software.
Omega Energy (previously called **Special Energy**). A shared pool of Energy, one for each side in a battle, which is primarily used to Upgrade or Awaken a Bot on that team. It is typically obtained by successfully using Software.

Condition. Shock, Rust, Overload, Freeze, Trapped, Draw Fire, or Stasis. Any Bot can be affected by any number of these conditions simultaneously. Rust and Overload are retained outside of battle, while the rest automatically go away at the end of a battle. When a Bot is healed by a Gearhead, its Endurance/Energy are both restored, and any conditions are removed.

Botlink. The menus to view the stats/status of the Bots on the player's team. There is a Botlink accessible from the pause menu and a Botlink accessible inside battle.

Medal. A special item earned for various achievements in the game, including Promotion Battles. Each Medal has a passive effect, some of which apply in battle. The player can only wear one Medal at a time, and only the effect of the worn Medal is active.

Power. The strength of a piece of Software, which determines how much damage it inflicts, if any

Energy Cost. The amount of energy that is consumed in order to use a piece of Software

Contact Effect. An effect (from an Ability or a piece of Hardware) that is triggered when an opposing Bot comes into contact with the Bot it affects.

Rank Formula

$$Experience_{min} = (Rank - 1)^3$$

Where:

$Experience_{min}$ = minimum experience required to level up to the given Rank

---

RAM Formula

TBD

Load Formula

TBD

Max-Endurance Formula

$$Endurance_{max} = \left[\frac{Endurance_{base} \times Rank}{10}\right]$$

Where:

$Endurance_{base}$ = base Endurance stat of Bot's current Blueprint

$Rank$ = Current Rank of Bot's Meister

Max-Energy Formula

$$Energy_{max} = \left[\frac{Energy_{base} \times Rank}{5}\right]$$

Where:

$Energy_{base}$ = base Energy stat of Bot's current Blueprint

$Rank$ = Current Rank of Bot's Meister

Current-Attack Formula
**applies to each Attack stat: Melee, Beam, and Blast*

If $Attack_{mod} \geq 0$

$$Attack_{current} = round_{up}\left[\left(1 + \frac{|Attack_{mod}|}{2}\right) \times Rank \times Attack_{base}\right]$$

If $Attack_{mod} < 0$

$$Attack = round_{up}\left[\left(1 + \frac{|Attack_{mod}|}{2}\right)^{-1} \times Rank \times Attack_{base}\right]$$

Where:

$Attack_{mod}$ = Bot's modifier stage for the relevant Attack stat (between -5 and 5 for each of Melee Attack, Beam Attack, and Blast Attack)

$Rank$ = Current Rank of Bot's Meister

$Attack_{base}$ = relevant base Attack stat of Bot's current Blueprint

---

Current-Defense Formula
**applies to each Defense stat: Melee, Beam, and Blast*

If $Defense_{mod} \geq 0$

$$Defense_{current} = round_{up}\left[\left(1 + \frac{|Defense_{mod}|}{2}\right) \times Rank \times Defense_{base}\right]$$

If $Defense_{mod} < 0$

$$Defense_{current} = round_{up}\left[\left(1 + \frac{|Defense_{mod}|}{2}\right)^{-1} \times Rank \times Defense_{base}\right]$$

Where:

$Defense_{mod}$ = Bot's modifier stage for the relevant Defense stat (between -5 and 5 for each of Melee Defense, Beam Defense, and Blast Defense)

$Rank$ = Current Rank of Bot's Meister

$Defense_{base}$ = relevant base Defense stat of Bot's current Blueprint

---

Current-Speed Formula

If $Speed_{mod} \geq 0$

$$Speed_{current} = \left(1 + \frac{|Speed_{mod}|}{2}\right) \times Rank \times Speed_{base}$$

If $Speed_{mod} < 0$

$$Speed_{current} = \left(1 + \frac{|Speed_{mod}|}{2}\right)^{-1} \times Rank \times Speed_{base}$$

Where:

$Speed_{mod}$ = Bot's Speed modifier stage (between -5 and 5)

$Rank$ = Current Rank of Bot's Meister

$Speed_{base}$ = base Speed stat of Bot's current Blueprint

Escape Formula

$$Threshold_{flee} = round_{up}\left[\frac{Speed_u}{\left[\frac{\sum Speed_o}{n}\right]} \times 100\right]$$

Where:

$Speed_u$ = user's Current Speed stat

$\sum Speed_o$ = the sum of each online opponent's current Speed stat

n = the number of online opponents

After calculating the flee threshold, choose a random integer between 0 and 100. If the integer is less than or equal to the flee threshold, user successfully escapes. If the integer is greater than the flee threshold, user fails to escape.

Hit-Check Formula

$$Accuracy_{mod} = Accuracy_u - Evasiveness_t$$

If $Accuracy_{mod} \geq 0$

$$Threshold_{hi} = round_{up}\left[\left(1 + \frac{|Accuracy_{mod}|}{3}\right) \times Accuracy_{Software}\right]$$

If $Accuracy_{mod} < 0$

$$Threshold_{hit} = round_{up}\left[\left(1 + \frac{|Accuracy_{mod}|}{3}\right)^{-1} \times Accuracy_{Software}\right]$$

Where:

$Accuracy_u$ = user's accuracy modifier stage (between -5 and 5)

$Evasiveness_t$ = target's evasiveness modifier stage (between -5 and 5)

$Accuracy_{Software}$ = base accuracy of the Software

After calculating the hit threshold, choose a random integer between 0 and 100. If the integer is less than or equal to the hit threshold, the Software hits. If the integer is greater than the hit threshold, the Software misses.

Critical-Hit Formula

$$Threshold_{crit} = round_{up}\left[Crit_{mod} \times Crit_{Software}\right]$$

Where:

$Crit_{mod}$ = the total effective crit-rate modifier—I haven't yet decided how I want to calculate this.

$Crit_{Software}$ = base critical-hit rate of the Software (usually 5, i.e., 5%)

After calculating the crit threshold, choose a random integer between 0 and 100. If the integer is less than or equal to the crit threshold, the Software lands a critical hit on the target. If the integer is greater than the crit threshold, the Software lands a regular hit on the target.

Damage Formula

$$Damage = round_{up}\left[\frac{\left(\frac{Attack_u}{Defense_t}\right) \times Power_{Software} \times c \times v}{2}\right]$$

Where:

$Attack_u$ = the relevant Current Attack stat of the user—this is determined by the Software's Aura type

$Defense_t$ = the relevant Current Defense stat of the target—this is usually determined by the Software's Aura type, with 3 exceptions that use a Current Defense stat that is different from the Software's Aura type

$Power_{Software}$ = base Power of the Software used

$c =$	1	if the hit is a regular hit
	2	if the hit is a critical hit
$v =$	1.5	if the user's Aura has the advantage against the target's Aura
	0.5	if the user's Aura has the disadvantage against the target's Aura
	1.0	if the user's Aura has neither the advantage nor the disadvantage against the target's Aura.

After calculating the damage done, it is deducted from the target's Current Endurance.

Omega-Energy Formula

$$Energy_{\omega} = w \times v \times c \times a$$

Where:

$Energy_{\omega}$ = the amount of Omega Energy gained from the Bot's move

$w =$	0	if the move was an Item, the move missed (incl. because the target was offline), the move failed (incl. because the user failed to escape), or the move was Software that consumes Omega Energy
	10	if the move succeeded and was Technical, Heal, or Guard Software
	15	if the move succeeded and was "Focus"
	20	if the move hit and was combat Software (Melee, Beam, or Blast) (probably going to change this eventually to be calculated based on the damage dealt)
$v =$	1.5	if the move had the advantage
	0.5	if the move had the disadvantage
	1.0	if the move had neither the advantage nor the disadvantage
$c =$	2.0	if the move landed a critical hit
	1.0	If the move landed a regular hit
$a =$	2.0	if the move belonged to the same Aura as the user's Aura when it used the move (e.g., if the user already had a Melee Aura when it used Melee Software)
	1.0	if the move belonged to a different Aura from the user's Aura when it used the move

---

Experience-Gain Formula

$$Yield_o = round_{up}\left[\left(\frac{l \times Rank_o}{5}\right) \times m \times \left(\frac{(2 \times Rank_o) + 10}{Rank_o + Rank_p + 10}\right)^{2.5}\right]$$

$$Experience_{rank} = \sum Yield_o$$

Where:

$Yield_o$ = Rank experience yielded by each opponent Bot defeated in the battle

$Rank_o$ = Rank of the defeated opponent Bot

$Rank_p$ = player's Current Rank

$l =$	60	if the defeated opponent Bot was in its Basic form
	120	if the defeated opponent Bot was in its Combat form
	240	if the defeated opponent Bot was in its Awakened form

$m =$ 1.0 if the defeated opponent Bot was a Rogue
1.5 if the defeated opponent Bot belonged to a generic challenger Meister
2.0 if the defeated opponent Bot belonged to a unique Meister

$Experience_{rank}$ = total Rank experience gained from the battle

---

Blueprint-Experience Formula
**calculated for each Bot on the player's side that is online after a victory, unless the Experience for the Bot's Blueprint is already maxed out at 1000.*

$$Yield_o = round_{up}[f \times m]$$

$$Experience_{Blueprint} = \sum Yield_o$$

Where:

$Yield_o$ = Blueprint experience yielded by each opponent Bot defeated in the battle

$f =$ 1 if the defeated opponent Bot was in its Basic form
2 if the defeated opponent Bot was in its Combat form
3 if the defeated opponent Bot was in its Awakened form

$m =$ 4 if the defeated opponent Bot was a Rogue
6 if the defeated opponent Bot belonged to a generic challenger Meister
10 if the defeated opponent Bot belonged to a unique Meister

$Experience_{Blueprint}$ = total Blueprint experience gained from the battle